(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number 11282674 A

(43) Date of publication of application: 15.10.99

(51) Int. Cl G06F 9/30
G06F 9/30
G06F 9/32
G06F 9/38
G06F 9/38

(21) Application number 10083368

(22) Date of filing: 30.03.98

(71) Applicant: MATSUSHITA ELECTRIC IND CO LTD

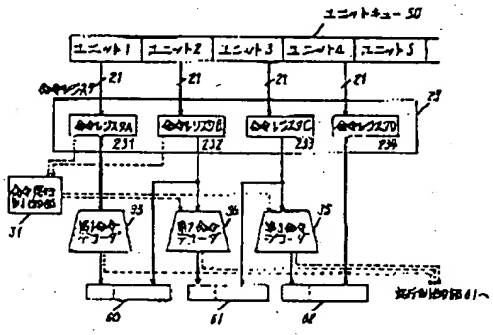
(72) Inventor HEIJI TAKEHITO
TANAKA TETSUYA
HIGAKI NOBUO
TAKAYAMA SHUICHI
KOTANI KENSUKE

(54) PROCESSOR

(57) Abstract:

PROBLEM TO BE SOLVED: To reduce the code size by using one or plural word elements to form a unit instruction serving as an execution unit and using two types of these word elements.

SOLUTION: An instruction issue control part 31 controls an instruction decoder by making reference to the format information as well as the parallel execution boundary information on the units which are stored in the instruction registers A231 and B232. Each of instruction decoders 33 to 35 inputs a unit of 21 bits to decode it, outputs the control signal concerning the operation of an instruction consisting of the said unit to an execution control part 41 and also outputs a constant operand included in the unit. In this case, the specific one of instruction registers 23 to which a certain unit is transferred is uniquely decided based on the position (order) of a unit queue 50. In other words, the units 1 and 2 are transferred to the registers A231 and B232 respectively Thus, a selector of units is not required.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－282674

(43)公開日　平成11年(1999)10月15日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	9/30	３５０	Ｇ０６Ｆ	9/30	３５０Ｆ
		３１０			３１０Ｃ
	9/32	３５０		9/32	３５０Ａ
	9/38	３１０		9/38	３１０Ｈ
		３７０			３７０Ｘ

審査請求　未請求　請求項の数12　ＯＬ　（全 19 頁）

(21)出願番号　特願平10－83368

(22)出願日　平成10年(1998)３月30日

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地

(72)発明者　瓶子　岳人
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　田中　哲也
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　檜垣　信生
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(74)代理人　弁理士　滝本　智之　（外１名）

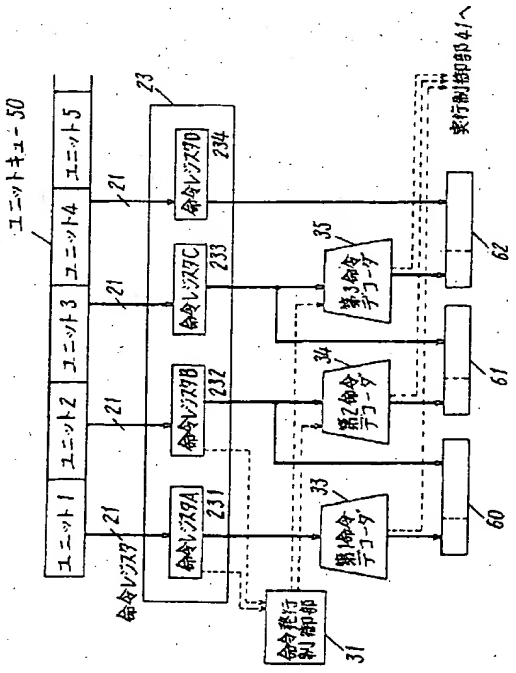
最終頁に続く

(54)【発明の名称】　プロセッサ

(57)【要約】

【課題】　ＶＬＩＷ方式のような静的な並列スケジューリングを前提としたプロセッサにおいて、コードサイズの削減を図る。また、可変長命令方式におけるハードウェア複雑化の問題点を克服し、性能向上とコード効率向上の両立を図る。

【解決手段】　命令構成要素内にフォーマット情報を持ち、それを参照することによりいくつの命令構成要素を連結して一つの命令を構成するのかを決定する。ハードウェア簡単化のため、命令構成要素内のフォーマット情報を参照することにより、後続する命令構成要素の命令としての解読を無効化する機構を備える。



【特許請求の範囲】
【請求項１】　一語が複数でかつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、
実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とするプロセッサ。
【請求項２】　前記単位命令を構成する前記語素の数に関する語数情報が、前記単位命令内に明示的に含まれることを特徴とする請求項１記載のプロセッサ。
【請求項３】　一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、
記憶装置から前記複合命令を読み出す複合命令読み出し手段と、
前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれる前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記単位命令を組み立てて解読する解読手段とを備えることを特徴とするプロセッサ。
【請求項４】　前記解読手段は、
前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、
前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする請求項３記載のプロセッサ。
【請求項５】　前記解読制御手段は、前記語数情報に基づいて、命令中のオペレーションに関する記述を解読していない前記単位解読手段に対して前記無効化を行うことを特徴とする請求項４記載のプロセッサ。
【請求項６】　前記一定数は、前記単位命令を構成する前記語素の最小の数であることを特徴とする請求項５記載のプロセッサ。
【請求項７】　前記プロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、
さらに、前記複合命令が一語に含み得る前記単位命令の数に等しい数の、前記単位命令を実行する実行手段を備えることを特徴とする請求項３から６のいずれか１項に記載のプロセッサ。
【請求項８】　前記単位命令は、第１の数の前記語素から構成される単位命令と、前記第１の数より大きい第２の数の前記語素から構成される単位命令とがあり、
前記第２の数の前記語素から構成される単位命令の内にあって、前記第１の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする請求項１から７のいずれか１項に記載のプロセッサ。
【請求項９】　前記複合命令中の前記単位命令と該単位命令に後続する単位命令とが並列に実行できるか否かを示す並列実行情報が、前記複合命令内に明示的に含まれることを特徴とする請求項１または２記載のプロセッサ。
【請求項１０】　前記並列実行情報が、並列に実行できる前記単位命令の群と後続する前記群との境界に関する情報であることを特徴とする請求項９記載のプロセッサ。
【請求項１１】　前記プロセッサは、
記憶装置から前記複合命令を読み出す複合命令読み出し手段と、
前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、前記語数情報と前記並列実行情報とに基づいて前記命令単位を組み立てて解読する解読手段とを備えることを特徴とする請求項９または１０記載のプロセッサ。
【請求項１２】　前記解読手段は、
前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、
前記語数情報と前記並列実行情報とに基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする請求項１１記載のプロセッサ。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、プロセッサに関し、特に並列処理による実行サイクル数の削減とコード効率の向上を図る技術に関する。
【０００２】
【従来の技術】近年のマイクロプロセッサ応用製品の高機能化及び高速化に伴い、高い処理性能を持つマイクロプロセッサ（以下、単に「プロセッサ」という。）が望まれている。そして、これを実現する技術の１つとして１サイクルに複数の命令を同時に実行するものがあり、ＶＬＩＷ（Very Long Instruction Word）方式のプロセッサもその技術の１つである。
【０００３】このＶＬＩＷ方式のプロセッサは、実行コード生成時にコンパイラ等により静的に命令間の依存関係を解析し、命令コードの移動を行って実行効率の良い命令ストリームを生成するものである。この方式は、動的に命令間の依存関係を解析するスーパースカラ方式のプロセッサと比べてハードウェアを簡略化でき、このため動作周波数を上げやすいという長所を有する。
【０００４】しかし、ＶＬＩＷ方式では、一般に命令を固定長として取り扱うため、次のような問題がある。
【０００５】すなわち、長い定数を扱う命令の指定には多くのビット数を必要とするが、レジスタ間演算命令の指定にはそれほど多くのビット数を必要としない等、命令毎に指定に必要なビット数にばらつきがある。ところが、ＶＬＩＷ方式では命令を固定長として取り扱うた

め、短いビット数しか必要としない命令を指定するのに必要以上のビット数を用いざるを得ず、コードサイズが大きくなってしまう。

【０００６】そして、この問題を解決する１つの手段として、命令長を可変とすることが考えられる。

【０００７】図１３は、１つの命令が１個または２個の命令構成要素（ここでは「ユニット」と呼ぶ）にて構成され、３つの命令を同時実行可能なプロセッサの命令レジスタ周辺の構成を示すブロック図である。同図において、破線は制御信号を表している。図１３においてユニットキュー５０は、ユニットの並びであり、命令メモリ等から供給された順にユニットを命令レジスタに転送していく。

【０００８】この構成では、命令レジスタＡ５２ａと命令レジスタＢ５２ｂ、命令レジスタＣ５２ｃと命令レジスタＤ５２ｄ、命令レジスタＥ５２ｅと命令レジスタＦ５２ｆがそれぞれ対になっており、命令は常に命令レジスタＡ５２ａ、命令レジスタＣ５２ｃ又は命令レジスタＥ５２ｅの３つのレジスタのいずれかを先頭として命令レジスタに格納され、２つのユニットを連結して１つの命令を構成する場合にのみ、対となっているもう一方の命令レジスタにユニットが転送される。したがって、命令レジスタＡ５２ａに転送されたユニットがそのユニット単体で命令を構成する場合には、命令レジスタＢ５２ｂにはユニットが転送されないことになる。

【０００９】図１３を見るとわかるように、この構成ではユニットキュー５０の各ユニットがいずれの命令レジスタに転送されるのかが一意に決まっていない。また、各命令レジスタへ転送されるユニットがユニットキュー５０のいずれのユニットなのかが一意に決まっていない。そこで、セレクタ５１ａ～５１ｄを制御して転送するユニットを選択することになる。さらに、これらのセレクタの制御は全体を一度に決定することができず、まずセレクタ５１ａ、セレクタ５１ｂの制御が決定され、命令レジスタＣへ転送されるユニットが決定してから、このユニット内の命令長に関する情報を参照して、図中破線で示すようにセレクタ５１ｃ、セレクタ５１ｄの制御を決定する。

【００１０】

【発明が解決しようとする課題】しかしながら、上記従来のプロセッサでは、ユニットキューから命令レジスタへの転送の際の遅延が大幅に大きくなるという問題点があった。これは、先行する命令レジスタに転送されたユニット内の命令長に関する情報を参照しなければ、当該命令レジスタに関するセレクタの制御を決定できないからである。また、並列度が増すに従って、転送すべき命令レジスタの数が増加するので、この遅延はさらに大きくなっていく。

【００１１】一方、ユニットキュー内のユニットと命令レジスタとの対応を一対一とし、図１３に示したプロセッサで問題となっていたユニットキューから命令レジスタへの転送におけるセレクタによる遅延の問題を解消する技術として、図１４に示すものがある。このプロセッサでは、命令を構成する可能性のある全てのユニットの組み合わせについてデコードしておき、先行する命令デコーダから出力される命令長の情報によって、デコード結果を選択して使用する。具体的には、図中破線で示すように、第１命令デコーダ５３ｄが出力する情報によりセレクタ５１ｅの制御を決定し、上記情報と第２命令デコーダ５３ｅまたは第３命令デコーダ５３ｆが出力する情報によりセレクタ５１ｆの制御を決定する。

【００１２】ところが、３つの命令を同時実行するために、２ユニット長の命令を解読するデコーダを５つも必要とし、ハードウェアが非常に大きくなるという問題点がある。

【００１３】そこで、本発明はかかる問題点を鑑みてなされたものであり、命令レベルの並列実行に際して、ハードウェア複雑化の問題を克服しつつ、性能向上とコード効率向上を両立するプロセッサを提供することを目的とする。

【００１４】

【課題を解決するための手段】上記目的を達成するために請求項１記載のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とする。

【００１５】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる。

【００１６】ここで請求項２記載のプロセッサは、請求項１記載のプロセッサにおいて、前記単位命令を構成する前記語素の数に関する語数情報が、前記単位命令内に明示的に含まれることを特徴とする。

【００１７】これによって、命令の発行制御部において、命令長を特定することが容易になり、命令発行制御を行うハードウェアを単純化することができる。

【００１８】また上記目的を達成するために請求項３記載のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれる前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記単位命令を組み立てて解読する解読手段とを備えることを特徴とする。

【００１９】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる

【００２０】また請求項４記載のプロセッサは、請求項３記載のプロセッサにおいて、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【００２１】これによって、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下するという問題点を克服している。

【００２２】また請求項５記載のプロセッサは、請求項１記載のプロセッサにおいて、前記解読制御手段は、前記語数情報に基づいて、命令中のオペレーションに関する記述を解読していない前記単位解読手段に対して前記無効化を行うことを特徴とする。

【００２３】これによって、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下する、という問題点を克服している。

【００２４】さらに請求項６記載のプロセッサは、請求項５記載のプロセッサにおいて、前記一定数は、前記単位命令を構成する前記語素の最小の数であることを特徴とする。

【００２５】また請求項７記載のプロセッサは、請求項３から請求項６のいずれか１項に記載のプロセッサにおいて、一語が複数個かつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、さらに、前記複合命令が一語に含み得る前記単位命令の数に等しい数の、前記単位命令を実行する実行手段を備えることを特徴とする

【００２６】また請求項８記載のプロセッサは、請求項１から請求項７のいずれか１項に記載のプロセッサにおいて、前記単位命令は、第１の数の前記語素から構成される単位命令と、前記第１の数より大きい第２の数の前記語素から構成される単位命令とがあり、前記第２の数の前記語素から構成される単位命令の内にあって、前記第１の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする。

【００２７】これによって、命令解読器の入力ビット幅を短くすることができ、ハードウェアを単純化することができる。

【００２８】さらに請求項９記載のプロセッサは、請求項１または請求項２記載のプロセッサにおいて、前記複合命令中の前記単位命令と該単位命令に後続する単位命令とが並列に実行できるかどうかを示す並列実行情報が、前記複合命令内に明示的に含まれることを特徴とする。

【００２９】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のＶＬＩＷなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【００３０】また請求項１０記載のプロセッサは、請求項９記載のプロセッサにおいて、前記並列実行情報が、並列に実行できる前記単位命令の群と後続する前記群との境界に関する情報であることを特徴とする。

【００３１】さらに請求項１１記載のプロセッサは、請求項９または請求項１０記載のプロセッサにおいて、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読み出し手段により読み出された前記複合命令中の前記語素から、前記語数情報と前記並列実行情報とに基づいて前記命令単位を組み立てて解読する解読手段とを備えることを特徴とする。

【００３２】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のＶＬＩＷなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【００３３】また請求項１２記載のプロセッサは、請求項１１記載のプロセッサにおいて、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報と前記並列実行情報とに基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【００３４】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のＶＬＩＷなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。さらに、命令の発行部のハードウェアを単純化することができ、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下するという問題点を克服している。

【００３５】

【発明の実施の形態】以下、本発明に係るプロセッサの実施の形態について、図面を用いて詳細に説明する。

（命令フォーマットとアーキテクチャの概要）まず、本プロセッサが解読実行する命令（特許請求の範囲に記載する「単位命令」に相当する。）の構造について説明する。

【００３６】図１（ａ）～図１（ｅ）は本プロセッサの命令フォーマットを示す図である。本プロセッサの各命令は、２１ビットの命令構成要素（ここでは「ユニッ

ト」と呼ぶ。特許請求の範囲に記載する「語素」に相当する。）にて構成されており、ユニット１つで構成される２１ビット命令とユニット２つで構成される４２ビット命令の２種類の命令フォーマットが存在する。各命令がいずれの長さの命令であるかは、１ビットのフォーマット情報１１によって決定される。具体的には、フォーマット情報１１が“０”の時はそのユニット単体で２１ビット命令となり、フォーマット情報１１が“１”の時はそのユニットとそれに後続するユニットとを連結して４２ビット命令となる。

【００３７】また、各命令には１ビットの並列実行境界情報１０を持たせてある。この情報は、この命令とそれに後続する命令との間に並列実行の境界が存在するか否かを示すものである。具体的には、並列実行境界情報１０が“１”の時はその命令と後続命令の間に並列実行の境界が存在し、並列実行境界情報１０が“０”の時には並列実行の境界が存在しないことになる。この情報の利用方法については後で述べる。

【００３８】各命令の命令長からフォーマット情報１１と並列実行境界情報１０を除いた部分にてオペレーションを指定する。２１ビット命令では１９ビット、４２ビット命令では４０ビットの長さを使用することができることになる。具体的には、“Ｏｐ１”、“Ｏｐ２”、“Ｏｐ３”のフィールドでは、オペレーションの種類を表すオペコードを、“Ｒｓ”のフィールドでは、ソースオペランドとなるレジスタのレジスタ番号を、“Ｒｄ”のフィールドでは、デスティネーションオペランドとなるレジスタのレジスタ番号を指定する。また、“ｉｍｍ５”及び“ｉｍｍ３２”のフィールドでは、それぞれ５ビットと３２ビットの演算用定数オペランドを指定する。そして、“ｄｉｓｐ１３”及び“ｄｉｓｐ３１”のフィールドでは、それぞれ１３ビットと３１ビットの変位（ディスプレースメント）を指定する。

【００３９】３２ビットの定数などの長い定数を扱う転送命令や演算命令、大きなディスプレースメントを指定する分岐命令は４２ビット命令で定義され、それらを除くほとんどの命令は２１ビット命令で定義されている。なお、図１を見てわかるように、４２ビット命令の構成要素である２つのユニットのうち、後ろの方のユニットすなわち２番目のユニットには、長い定数やディスプレースメントの一部のみが配置され、オペコードは配置されない。

【００４０】次に、本プロセッサのアーキテクチャの概要について説明する。本プロセッサは、静的な並列スケジューリングを前提としたプロセッサであって、命令の供給と発行の概念は図２のようになる。

【００４１】命令の供給は、同図（ａ）に示すように毎サイクル６４ビット固定長の命令供給単位（ここでは「パケット」と呼ぶ。特許請求の範囲に記載する「複合命令」に相当する。）でユニットを３個ずつ供給する。ユニット３個分の長さは６３ビットであるが、残りの１ビットについては使用しない。そして、命令の実行は、同図（ｂ）に示すように１サイクルで並列実行の境界までのユニット（ここでは「実行単位」と呼ぶ）を同時実行する。つまり、各サイクルにおいて並列実行境界情報１０が“１”である命令までの命令を並列実行することになる。供給されながら実行されずに残ったユニットは、命令バッファに蓄積され、次のサイクル以降で実行の対象となる。

【００４２】つまり、このアーキテクチャでは、固定長のパケット単位で命令を供給しておき、静的に求めた情報を元に、各サイクルにおいて並列度に応じた適切な数のユニットを発行していく、ということになる。この手法をとることにより、通常の固定長命令のＶＬＩＷ方式で発生していた無動作命令（ｎｏｐ命令）が全く無くなり、コードサイズを削減することができる。

【００４３】また、命令内のフォーマット情報１１の値によって、２つのユニットを１命令として実行する場合と１つのユニットを１命令として実行する場合がある。この手法をとることにより、命令の指定に多くのビット数を必要とする一部の命令に対してのみ長い命令フォーマットを使用し、他のほとんどの命令については短い命令フォーマットで指定することができるので、さらにコードサイズを削減することができる。具体例については後に述べる。

（プロセッサのハードウェア構成）次に、本プロセッサのハードウェア構成を説明する。

【００４４】図３は、本発明に係るプロセッサのハードウェア構成を示すブロック図である。

【００４５】本プロセッサは、１サイクルに最大３つの命令を並列実行するプロセッサであり、大きく分けて、命令供給発行部２０、解読部３０、実行部４０から構成される。

【００４６】命令供給発行部２０は、図示されていない外部メモリから命令群を供給し、解読部３０に出力するものであり、命令フェッチ部２１、命令バッファ２２及び命令レジスタ２３からなる。

【００４７】命令フェッチ部２１は、３２ビットのＩＡ（インストラクションアドレス）バス及び６４ビットのＩＤ（インストラクションデータ）バスを通じて図示されていない外部メモリからユニットのブロックをフェッチし、内部の命令キャッシュに保持すると共に、ＰＣ部４２から出力されたアドレスに相当するユニット群を命令バッファ２２に供給する。

【００４８】命令バッファ２２は、６３ビットのバッファを２個備えており、命令フェッチ部２１によって供給されたユニットを蓄積しておくために用いられる。命令バッファ２２へは、命令フェッチ部２１から６４ビット単位でパケットが供給される。ここで、パケットの最上位の１ビットの情報は使用されない。命令バッファ２２

に蓄積されたユニットは、命令レジスタ２３の適切なレジスタに出力される。なお、命令バッファ２２については、別の図面においてさらに詳細な構成を示している。

【００４９】命令レジスタ２３は、４個の２１ビットレジスタからなり、命令バッファ２２から送られてきたユニットを保持するためのものである。命令レジスタ２３周辺については、別の図面においてさらに詳細な構成を示している。

【００５０】解読部３０は、命令レジスタ２３に保持された命令を解読し、その解読結果に応じた制御信号を実行部４０に出力するものであり、大きく分けて、命令発行制御部３１と命令デコーダ３２からなる。

【００５１】命令発行制御部３１は、命令レジスタ２３の１個のレジスタに保持されたユニットに対して、ユニット内の並列実行境界情報１０とフォーマット情報１１を参照することによって、２つのユニットを１つの命令として扱うように制御したり、並列実行の境界を越えたユニットについては、そのユニットの発行を無効化したりといった発行に関する制御を行う。なお、命令発行制御部３１については、別の図面においてさらに詳細な動作説明を行う。

【００５２】命令デコーダ３２は、命令レジスタ２３に格納された命令群を解読する装置であり、第１命令デコーダ３３、第２命令デコーダ３４及び第３命令デコーダ３５からなる。これらのデコーダは、基本的に１サイクルに１つの命令を解読し、実行部４０に制御信号を与える。また、命令内に置かれた定数オペランドについては、各命令デコーダから実行部４０のデータバス４８に転送される。

【００５３】実行部４０は、解読部３０での解読結果に基づいて、最大３つの命令を並列実行する回路ユニットであり、実行制御部４１、ＰＣ部４２、レジスタファイル４３、第１演算部４４、第２演算部４５、第３演算部４６、オペランドアクセス部４７及びデータバス４８、４９からなる。

【００５４】実行制御部４１は、解読部３０での解読結果に基づいて実行部４０の各構成要素４２～４９を制御する制御回路や配線の総称であり、タイミング制御、動作許可禁止制御、ステータス管理、割り込み制御等の回路を有する。

【００５５】ＰＣ（プログラムカウンタ）部４２は、次に解読実行すべき命令が置かれている図示されていない外部メモリ上のアドレスを命令供給発行部２０の命令フェッチ部２１に出力する。

【００５６】レジスタファイル４３は、Ｒ０～Ｒ３１の３２個の３２ビットレジスタから構成される。これらのレジスタに格納された値は、第１命令デコーダ３３、第２命令デコーダ３４及び第３命令デコーダ３５での解読結果に基づいて、データバス４８を経由して第１演算部４４、第２演算部４５及び第３演算部４６に転送され、そこで演算が施され、又はそこを単に通過した後に、データバス４９を経由してレジスタファイル４３またはオペランドアクセス部４７に送られる。

【００５７】第１演算部４４、第２演算部４５及び第３演算部４６は、それぞれ２個の３２ビットデータに対して算術論理演算を行うＡＬＵや乗算器と、シフト演算を行うバレルシフタを内部に有し、実行制御部４１による制御の下で演算を実行する。

【００５８】オペランドアクセス部４７は、レジスタファイル４３と図示されていない外部メモリとの間でオペランドの転送を行う回路である。具体的には、例えば、命令内で、オペコードとして“ｌｄ”（ロード）が置かれていた場合には、外部メモリに置かれていた１ワード（３２ビット）のデータがオペランドアクセス部４７を経てレジスタファイル４３の指定されたレジスタにロードされ、また、オペコードとして“ｓｔ”（ストア）が置かれていた場合には、レジスタファイル４３の指定されたレジスタの格納値が外部メモリにストアされる。

【００５９】上記ＰＣ部４２、レジスタファイル４３、第１演算部４４、第２演算部４５、第３演算部４６及びオペランドアクセス部４７は、図示されるように、データバス４８（Ｌ１バス、Ｒ１バス、Ｌ２バス、Ｒ２バス、Ｌ３バス、Ｒ３バス）及びデータバス４９（Ｄ１バス、Ｄ２バス、Ｄ３バス）で接続されている。なお、Ｌ１バス及びＲ１バスはそれぞれ第１演算部４４の２つの入力ポートに、Ｌ２バス及びＲ２バスはそれぞれ第２演算部４５の２つの入力ポートに、Ｌ３バス及びＲ３バスはそれぞれ第３演算部４６の２つの入力ポートに、Ｄ１バス、Ｄ２バス及びＤ３バスはそれぞれ第１演算部４４、第２演算部４５及び第３演算部４６の出力ポートに接続されている。

（命令バッファの詳細な構成）次に、命令バッファ２２の詳細な構成を説明する。

【００６０】図４は、命令バッファ２２の詳細な構成を示すブロック図である。命令バッファ２２は命令バッファＡ２２１及び命令バッファＢ２２２の２個の６３ビットのバッファからなり、それぞれ３個ずつのユニットを保持することができる。命令バッファＡ２２１はバッファＡ０、Ａ１及びＡ２からなり、それぞれ１個ずつのユニットを保持することができる。同様に、命令バッファＢはバッファＢ０、Ｂ１及びＢ２からなる。

【００６１】命令バッファ２２には、命令フェッチ部２１から６４ビット単位でパケットが供給される。ただし、パケットの最上位の１ビットの情報は使用されない。この際、命令バッファＡ２２１と命令バッファＢ２２２にまたがって供給されることはなく、いずれかのバッファに６３ビット単位で供給されることになる。命令バッファ２２に蓄えられたユニットは供給された順序を保っており、その順序やいずれのバッファが有効であるかについては命令バッファ制御部２２３により、状態と

して管理されている。

【００６２】命令バッファ制御部２２３は、毎サイクルバッファ内の有効なユニットを順に命令レジスタ２３に転送するため、セレクタ２２４ａ～２２４ｄの制御を行う。この制御により、命令バッファ２２内の先頭の４つのユニットが命令レジスタ２３に転送される。さらに、命令レジスタ２３に転送したユニットの中でどれだけのユニットが発行されずに残ったか、という解読部３０の命令発行制御部３１からの情報と、命令フェッチ部２１から転送されてきたユニットの内いずれのユニットが有効かという情報とを元に、命令バッファ２２の状態の更新を行う。

【００６３】具体的には、まず命令バッファ２２が空の状態で、あるパケットの２番目のユニットに分岐した場合には、命令フェッチ部２１からそのパケットが供給され、供給されたパケットは命令バッファＡ２２１に転送される。そのパケットの先頭のユニットは無効なので、命令バッファ制御部２２３の制御により、命令バッファ２２の状態としてバッファＡ１及びバッファＡ２のみが有効な状態となる。

【００６４】次のサイクルで命令バッファ２２から命令レジスタ２３に転送したユニットが全く発行されず、命令フェッチ部２１から６４ビットの有効なパケットが供給された場合には、そのパケットは命令バッファＢ２２２に転送され、命令バッファ２２の状態は、バッファＡ１、Ａ２、Ｂ０、Ｂ１及びＢ２が有効な状態となる。

【００６５】さらに、次のサイクルでは、命令バッファ２２に空きがないので、命令フェッチ部２１からの供給は受け付けず、命令レジスタ２３へは、順にバッファＡ１、バッファＡ２、バッファＢ０、バッファＢ１のユニットを転送する。

【００６６】このように、命令バッファ２２に６３ビット単位で空きがある場合にのみ命令フェッチ部２１からパケットの供給を行い、供給された順を管理しておき、各サイクルにおいて、供給された順に先頭の４つのユニットを命令レジスタ２３に転送していく。

（命令レジスタ２３周辺の構成と命令発行制御部３１の動作）次に、命令レジスタ２３周辺の構成を示し、命令発行制御部３１の詳細な動作を説明する。

【００６７】図５は、命令レジスタ２３周辺の構成を示すブロック図である。図中、破線の矢印は制御信号を表す。

【００６８】命令レジスタ２３は命令レジスタＡ２３１、命令レジスタＢ２３２、命令レジスタＣ２３３及び命令レジスタＤ２３４の４個の２１ビットレジスタからなる。命令レジスタ２３には、命令バッファ２２からユニットが供給されるわけだが、わかりやすくするために命令バッファ２２から供給されるユニットの並びであるユニットキュー５０という概念を考える。そして、ここでは命令レジスタ２３にはユニットキュー５０からユニットが供給されると考える。

【００６９】図５を見るとわかるように、あるユニットがいずれの命令レジスタ２３に転送されるかどうかは、ユニットキュー５０での位置（順序）によって一意に決まる。つまり、ユニット１は命令レジスタＡ２３１へ、ユニット２は命令レジスタＢ２３２へ転送されることになる。これにより、ユニットキュー５０から命令レジスタ２３への転送を行う際に、図１３の従来例において存在したようなユニットの選択を行うセレクタが不要となり、ハードウェアが単純化されており、遅延も最小限に抑えられている。

【００７０】図中３３～３５の各命令デコーダは、２１ビットのユニットを入力とし、それを解読して、そのユニットが構成する命令の動作に関する制御信号を実行制御部４１に出力するとともに、ユニット内に配置された定数オペランドを出力する。図１の命令フォーマットからわかるように、４２ビット命令を構成する２つのユニットのうち、２番目のユニットには定数オペランドの一部しか配置されない。つまり、このユニットにはオペコードが存在しないため、命令デコーダに入力する必要がない。そこで、各命令の定数オペランドは、図５に示されるように、命令デコーダが出力したユニット内の定数と、命令レジスタから無条件に直接転送された定数とを連結したものということになる。図５の６０～６２が各命令の定数オペランドである。

【００７１】また、各命令デコーダには、制御信号として１ビットの無動作命令フラグが入力される。このフラグを“１”にセットすると、そのデコーダは出力として無動作命令を出力する。つまり、無動作命令フラグをセットすることにより、その命令デコーダの命令としてのデコードを無効化することができる。

【００７２】ここで、命令レジスタ２３に格納されたユニットを組み合わせて命令として発行する制御を行う命令発行制御部３１の動作について説明する。

【００７３】命令発行制御部３１は、命令レジスタＡ２３１及び命令レジスタＢ２３２に格納された各ユニットの並列実行境界情報１０とフォーマット情報１１を参照することにより命令デコーダの制御を行う。

【００７４】まず、これらの情報から、命令レジスタ２３に格納されたユニットの内どこまでをこのサイクルで発行するのかを求める。そして、どれだけのユニットが発行されずに残ったのかの情報を命令バッファ２２内の命令バッファ制御部２２３に伝達する。

【００７５】次に命令デコーダ３２を制御し、このサイクルで発行される命令についてのみ解読を行うように制御する。図５からわかるように、命令としてデコードされる可能性のあるユニットは、命令レジスタＡ２３１、命令レジスタＢ２３２及び命令レジスタＣ２３３に格納されたユニットのみである。そこで、ユニット内の情報を参照して、これらのユニットの中で、４２ビット命令

の2ユニット目にあたるものや発行されずに残るものに関しては、そのユニットの命令としてのデコードを無効化する。４２ビット命令の2ユニット目にあたるユニットは、直前のユニットが構成する命令の定数オペランドの一部として直接出力される。

【００７６】具体的には、命令レジスタＡ２３１のユニット（ユニット１）のフォーマット情報１１が“１”のときには、ユニット１と命令レジスタＢ２３２のユニット（ユニット２）とを連結して４２ビット命令となるので、ユニット２の命令としてのデコードを無効化する、すなわち第２命令デコーダ３４の無動作命令フラグを“１”にセットする。図５において、命令発行制御部３１から第２命令デコーダ３４への破線がこの動作に相当する。ユニット２は、ユニット１が構成する命令の定数オペランド６０の一部として直接出力される。

【００７７】また、ユニット１のフォーマット情報１１が“０”、ユニット２のフォーマット情報が“１”の時は、ユニット２と命令レジスタＣ２３３のユニット（ユニット３）とを連結して４２ビット命令となるので、ユニット３の命令としてのデコードをキャンセルする、すなわち第３命令デコーダ３５の無動作命令フラグを“１”にセットする。図５において、命令発行制御部３１から第３命令デコーダ３５への破線がこの動作に相当する。ユニット３は、ユニット２が構成する命令の定数オペランド６１の一部として直接出力される。

【００７８】このように、フォーマット情報１１を参照することにより、必要に応じて命令デコーダの無動作フラグを設定し、命令としてのデコードを無効化する。

【００７９】それから、ユニット１の並列実行境界情報１０が“１”、フォーマット情報１１が“０”のときは、このサイクルではユニット１までしか発行されないので、ユニット２とユニット３の命令としてのデコードを無効化する、すなわち第２命令デコーダ３４と第３命令デコーダ３５の無動作命令フラグを共に“１”にセットする。図５において、命令発行制御部３１から第２命令デコーダ３４と第３命令デコーダ３５への破線がこの動作に相当する。

【００８０】また、ユニット１の並列実行境界情報１０が“０”、ユニット２の並列実行境界情報１０が“１”、フォーマット情報１１が共に“０”のときは、このサイクルではユニット２までしか発行されないので、ユニット３の命令としてのデコードを無効化する、すなわち第３命令デコーダ３５の無動作命令フラグを共に“１”にセットする。図５において、命令発行制御部３１から第３命令デコーダ３５への破線がこの動作に相当する。

【００８１】このように、並列実行境界情報１０を参照することにより、必要に応じて命令デコーダの無動作フラグを設定し、命令としてのデコードを無効化する。

【００８２】以上のような命令発行制御を実現する命令発行制御部３１とその周辺回路の構成を図６に示す。

【００８３】前述のように命令発行制御部３１は命令レジスタＡ２３１及び命令レジスタＢ２３２に格納されたユニットの並列実行境界情報１０とフォーマット情報１１を参照し、第２命令デコーダ３４及び第３命令デコーダ３５の命令としてのデコードを無効化するかどうかを決定する無動作命令フラグとなる制御信号を出力する。

【００８４】図６のような回路構成をとることにより、第２命令デコーダ３４は、命令レジスタＡ２３１に格納されたユニットの並列実行境界情報１０が“１”であるか、またはそのユニットのフォーマット情報１１が“１”であるときに無効化される。また、第３命令デコーダ３５は、命令レジスタＡ２３１に格納されたユニットもしくは命令レジスタＢ２３２に格納されたユニットの並列実行境界情報１０が“１”であるか、または命令レジスタＢ２３２に格納されたユニットのフォーマット情報１１が“１”であるときに無効化される。

【００８５】このように、図１に示したような命令フォーマットをとり、図６に示したような単純な回路を用意するだけで、必要最低限の情報を参照して高速な命令発行制御を行うことができる。

【００８６】以上で述べたような命令発行制御の方法をとることにより、１サイクルで同時発行可能な命令の命令長の組み合わせに多少の制限が生じる。本プロセッサで同時発行可能な命令の命令長の組み合わせを図７に示す。

【００８７】図７を見るとわかるように、本プロセッサでは、ユニットの並びの先頭から３つ目までのユニットについてのみ命令としてデコードすることができる。つまり、図中（ａ）～（ｈ）のパターンについて発行することができる。最大で４つのユニットを同時に発行できることになる。ただし、４つのユニットを発行するパターンの内、図中（ｉ）、（ｊ）のパターンについては同時発行することができない。

（従来の命令発行制御方法との比較）ここで、本実施形態のプロセッサと本発明によらない従来のプロセッサとの比較を行う。

【００８８】まず、図１３に示した従来例において、ユニットキュー５０から命令レジスタへの転送において、セレクタによる遅延が問題となっていたが、本発明のプロセッサでは、ユニットキュー５０内のユニットと各命令レジスタが一対一に対応しているため、図１３において存在していたセレクタ５１ａ～５１ｄが不要となり、上記遅延の問題が解決されている。

【００８９】また、図１３の構成では、並列度が増して転送すべき命令レジスタが増加していくに従って、セレクタが増加し遅延がさらに大きくなっていくのに対して、本発明のプロセッサでは、ユニットキュー５０と命令レジスタの対応は一対一なので、遅延が大きくなることはない。

【００９０】一方、この可変長命令方式をスーパースカラ方式にて並列実行を行うプロセッサに適用したものも提案されている。例えば、論文 The Approach to MultipleInstruction Execution in the GMICRO/400 Processor （PROCEEDINGS, The Eighth TRON Project Symposium(International), 1991参照）にて開示されているＧＭＩＣＲＯ／４００がある。この技術は、図１４の概念をとりながらもハードウェアを削減するために制限を設けている。

【００９１】図１５は、ＧＭＩＣＲＯ／４００で採用されている命令発行制御方法をとった場合の命令レジスタ周辺の構成を示すブロック図である。図１５において、破線は制御信号を表し、５４ａ及び５４ｂは命令内に指定された定数オペランドを表す。命令デコーダは、入力された命令を解読し、その結果その命令の実行を制御する信号を実行制御部に出力すると共に、命令内に指定された定数オペランドを出力する。

【００９２】ＧＭＩＣＲＯ／４００の命令発行制御方法では、ユニット１とユニット２を連結したもの、ユニット２及びユニット３をそれぞれ一旦デコードしておき、第１命令デコーダ５３ｉの解読によって１番目の命令が１ユニット長の命令なのか２ユニット長の命令なのかが判明した時点で、セレクタ５１ｇおよびセレクタ５１ｈを制御することにより、第２命令デコーダ５３ｊもしくは第３命令デコーダ５３ｋの解読結果を選択して使用する。

【００９３】図１５を見るとわかるように、ＧＭＩＣＲＯ／４００では、図１４の構成に対して、同時実行可能な命令数を３から２に減らすことにより、第４命令デコーダ５３ｇと第５命令デコーダ５３ｈを削除している。また、第２命令デコーダ５３ｊと第３命令デコーダｋは、入力ビット幅を１ユニット長とし、ハードウェア削減を図っている。しかし、これによって、同時実行される２番目の命令は１ユニット長の命令のみという制限が発生する。

【００９４】以上のようなハードウェア削減を図っても、２命令同時発行を可能にするために３つの命令デコーダを必要としており、依然としてハードウェア量が多いという問題点がある。

【００９５】また、図１５の構成では、第１命令デコーダ５３ｉの解読が完了するまでセレクタ５１ｇ、５１ｈの制御を決定することができない。このセレクタの制御が決定するまで、２番目の命令として第２命令デコーダ５３ｊと第３命令デコーダ５３ｋのいずれの解読結果を用いるかを決定できず、オペランドとなるレジスタの格納値の読み出しを開始できない。オペランドとなる可能性のある全てのレジスタの格納値を先行的に読み出しておき、それを選択して使用する方法も考えられるが、レジスタファイルの読み出しポート数が増加するため実用的ではない。このように、図１５の構成では読み出すレジスタを確定するまでの遅延が大きくなる。実際、ＧＭＩＣＲＯ／４００では命令解読ステージの直後のステージでは演算の実行は行わず、オペランドを読み出すステージとしている。

【００９６】さらに、図１５の構成にて並列度が増して同時発行可能なユニット数が増加していくと、セレクタの数が増し、制御が複雑化するという問題点がある。

【００９７】以上に述べたように、スタティックスケジューリングによってさらなる並列化を実現し、性能向上を図ることができるが、コードサイズが大きくなるという問題点がある。また、コードサイズを削減する手段として可変長命令方式があるが、ハードウェアが複雑になるという問題点がある。

【００９８】そして、図１５に示したＧＭＩＣＲＯ／４００の例においては、命令デコーダを３つ用意しても最大２命令しが同時実行できないのに対して、本発明のプロセッサの命令発行制御方法を用いると、３つの命令デコーダにて最大３命令を同時実行することができる。逆に、本発明において、最大２命令を同時発行する構成を想定した場合、２個の命令デコーダにて構成することができる。具体的な命令レジスタ周辺の構成は図１６のようになる。これにより、ハードウェアを削減することができる。

【００９９】また、図１５の構成では、第１命令デコーダ５３ｉの解読が完了するまでセレクタ５１ｇ、５１ｈの制御を決定することができず、２番目の命令として第２命令デコーダ５３ｊと第３命令デコーダ５３ｋのいずれの解読結果を用いるかを決定できない。そのため、オペランドとなるレジスタを確定するまでの遅延が大きくなる。これに対して、本発明のプロセッサの構成では、他のデコーダの解読結果を待たずに、オペランドとなるレジスタを確定することができるため、解読ステージの前半にオペランドとなるレジスタの読み出しを開始することができる。その結果、解読ステージの完了時点で、オペランドとなるレジスタの読み出しも完了させておくことができる。これによって、解読ステージの直後のステージで演算を実行することができ、実行効率を高めることができる。

【０１００】さらに、本発明の構成では、同時発行可能なユニット数が増加しても単純にデコーダの数を増していけばよいのに対して、図１５の構成では、同時発行可能なユニット数が増加していくと、セレクタの数が増して制御が複雑化するという問題点がある。

【０１０１】それから、命令フォーマットの違いによる差異として次のものがある。本発明では図１のように、２ユニット長の命令の２番目のユニットには定数オペランドの一部のみが配置されるため、図５のように２番目のユニットは命令デコーダには入力されずに直接オペランドとして出力される。このため、すべての命令デコーダは１ユニット長の命令を解読するだけでよい。これに

対して、ＧＭＩＣＲＯ／４００では、２ユニット長の命令の２番目のユニットにもオペコードが配置されるため、図１５の構成で第１命令デコーダ５３ｉは２ユニット長の命令を解読する必要があり、本発明の構成に比べてハードウェアが増加している。

（プロセッサの動作）次に、具体的な命令を解読実行した場合の本実施形態のプロセッサの動作について説明する。

【０１０２】図８は、３２ビットの定数を扱う処理の一例を示すフローチャートである。本図に示されている処理は、３２ビットの定数“０ｘ８７６５４３２１”をレジスタＲ１に転送し（ステップＳ１００）、レジスタＲ５の格納値をレジスタＲ０に転送し（ステップＳ１０１）、レジスタＲ０の格納値にレジスタＲ１の格納値を加え（ステップＳ１０２）、レジスタＲ３の格納値にレジスタＲ２の格納値を加え（ステップＳ１０３）、レジスタＲ０の格納値をメモリ内のレジスタＲ４の格納値で示されるアドレスに格納し（ステップＳ１０４）、レジスタＲ０の格納値をレジスタＲ６に転送し（ステップＳ１０５）、最後にレジスタＲ３の格納値をレジスタＲ７に転送する（ステップＳ１０６）というものである。

【０１０３】図９は、図８に示された処理を本プロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【０１０４】このプログラムは、７個の命令で構成されており、命令供給単位としては３個のパケット７０～７２から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。具体的には、ニーモニック“ｍｏｖ”は、定数及びレジスタの格納値のレジスタへの転送を表し、ニーモニック“ａｄｄ”は、定数及びレジスタの格納値とレジスタの格納値との加算を表し、ニーモニック“ｓｔ”は、レジスタの格納値のメモリへの転送を表している。

【０１０５】なお、定数は１６進数で表現されている。また、“Ｒｎ（ｎ＝０～３１）”はレジスタファイル４３の中の一つのレジスタを示す。そして、各命令の並列実行境界情報１０とフォーマット情報１１についても“０”又は“１”で示してある。

【０１０６】図９（ｂ）を用いて、図８に示された処理における各実行単位ごとの本プロセッサの動作を説明する。

（実行単位１）パケット７０がメモリから供給され、パケット７０内のユニットが順に命令レジスタ２３に転送される。次に、命令発行制御部３１が各ユニットの並列実行境界情報１０とフォーマット情報１１を参照して発行を制御する。具体的には、１番目のユニットのフォーマット情報１１が“１”であるので、１番目のユニットと２番目のユニットを連結して１つの命令として扱う。つまり、第２命令デコーダ３４の無動作命令フラグを“１”にセットして、命令としてのデコードを無効化する。また、１番目のユニットの並列実行境界情報１０が“０”であり、３番目のユニットの並列実行境界情報１０が“１”であるので、３番目のユニットまでの２個の命令を発行する。供給されたすべてのユニットを発行するため、命令バッファ２２にはユニットは蓄積されない。

【０１０７】実行部４０では、レジスタＲ１に定数“０ｘ８７６５４３２１”が転送され、レジスタＲ５の格納値がレジスタＲ０に転送される。

（実行単位２）パケット７１がメモリから供給され、パケット７１内のユニットが順に命令レジスタ２３に転送される。３個のユニット共フォーマット情報１１が“０”であるので、いずれのユニットも２１ビット命令となる。また、１番目のユニットの並列実行境界情報１０が“０”であり、２番目のユニットの並列実行境界情報１０が“１”であるので、２番目のユニットまでの２個の命令を発行する。３番目のユニットは、発行されずに残ったので命令バッファ２２に蓄積される。

【０１０８】実行部４０では、レジスタＲ０の格納値にレジスタＲ１の格納値が加えられてレジスタＲ０に格納され、レジスタＲ３の格納値にレジスタＲ２の格納値が加えられてレジスタＲ３に格納される。

（実行単位３）パケット７２がメモリから供給され、命令バッファ２２に蓄積されていた１個のユニットとパケット７２内の２個のユニットとが順に命令レジスタ２３に転送される。３個のユニット共フォーマット情報１１が“０”であるので、いずれのユニットも２１ビット命令となる。また、１番目のユニットの並列実行境界情報１０と２番目のユニットの並列実行境界情報が“０”であり、３番目のユニットの並列実行境界情報１０が“１”であるので、３番目のユニットまでの３個の命令を発行する。これで、供給されたユニットはすべて発行されたことになる。

【０１０９】実行部４０では、レジスタＲ０の格納値がメモリ内のレジスタＲ４の格納値で示されるアドレスに転送され、レジスタＲ０の格納値がレジスタＲ６に転送され、レジスタＲ３の格納値がレジスタＲ７に転送される。

【０１１０】以上のようにして、本プロセッサにおいて図８に示した処理を行うプログラムは３つの実行単位で実行される。実行コードは、４２ビット命令が１個と２１ビット命令が６個で構成されていたので、コードサイズは１６８ビットである。

（従来の固定長ＶＬＩＷ方式のプロセッサとの比較）次に、図８に示した処理を、従来技術の１つとして挙げた命令長が固定のＶＬＩＷ方式のプロセッサに行わせた場合を仮定して、本発明に係るプロセッサの場合と比較する。

【０１１１】毎サイクル固定長の命令を固定個数発行する単純なＶＬＩＷ方式では、３２ビットの定数を転送す

る命令を１命令で指定できるような命令長にすると、非常にコードサイズが大きくなってしまうため、命令長は３２ビットとし、３２ビットの定数の転送は１６ビットずつ２命令に分けて行うことにする。

【０１１２】図１０は、図８に示された処理を、命令長が３２ビット固定のＶＬＩＷ方式のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【０１１３】このプログラムは、４個のパケット７３～７６から構成されている。各命令の処理内容は、図９に示したコードと同様に、各フィールドに置かれたニーモニックで表現されている。ただし、ニーモニック“ｓｅｔｈｉ”は、１６ビットの定数をレジスタの上位１６ビットに格納することを表し、ニーモニック“ｓｅｔｌｏ”は、１６ビットの定数をレジスタの下位１６ビットに格納することを表し、ニーモニック“ｎｏｐ”は、何もしない命令であることを表している。

【０１１４】図１０（ａ）の実行コードと同図（ｂ）の実行イメージとを比較するとわかるように、ＶＬＩＷ方式では、各サイクル供給された命令がそのまま発行される。つまり、毎サイクル３２ビット命令が３個発行されることになる。並列実行可能な命令が存在しない場合は、あらかじめソフトウェアで“ｎｏｐ”命令を挿入しておく必要がある。そのため、この例でも４個の“ｎｏｐ”命令が挿入されて、コードサイズは３２ビット命令が１２個なので３８４ビットとなっており、本発明に係るプロセッサの場合のコードサイズよりも大幅に大きいものになっている。

【０１１５】また、３２ビットの定数のレジスタへの転送を２命令に分けて行っているために新たな依存関係が生じ、実行単位の数が４つとなっている。どのような命令並べ替えを行っても実行単位の数を減らすことはできない。これによって、本発明に係るプロセッサの場合に比べて実行サイクル数が１サイクル増加する。

（従来の並列実行境界の情報を固定長命令内に持つプロセッサとの比較）次に、図８に示した処理を、従来技術の１つとして挙げた命令長が固定であり並列実行の境界であるか否かの情報を命令内に持つ方式のプロセッサに行わせた場合を仮定して、本発明に係るプロセッサの場合と比較する。

【０１１６】この方式では、命令長が３２ビットのモデルと４０ビットのモデルを考える。命令長が３２ビットのモデルでは、図１０のＶＬＩＷ方式の場合と同様に、３２ビットの定数のレジスタへの転送は２命令に分けて行う。それに対して命令長が４０ビットのモデルでは、３２ビットの定数のレジスタへの転送を含むすべての種類の演算を１命令で指定することができる。

【０１１７】図１１は、図８に示された処理を、命令長が３２ビット固定であり並列実行の境界の情報を命令内に持つ方式のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【０１１８】このプログラムは、８個の命令で構成されており、命令供給単位としては３個のパケット７７～７９から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。３２ビットの定数のレジスタへの転送は、図１０の命令長３２ビット固定のＶＬＩＷ方式の場合と同様に１６ビットずつ２個の命令に分けて行う。

【０１１９】図１１を見ると分かるように、このモデルでも図１０のＶＬＩＷ方式の場合と同様に３２ビットの定数のレジスタへの転送を２命令に分けて実行しているため、新たな依存関係が生じ、実行サイクル数が本発明に係るプロセッサの場合に比べて１サイクル増加している。

【０１２０】コードサイズに関しては、“ｎｏｐ”命令の挿入が発生しないため、図１０のＶＬＩＷ方式の場合のコードサイズから丁度“ｎｏｐ”命令の分だけ削減されており、３２ビット命令が８個で２５６ビットとなっている。しかし、依然として本発明に係るプロセッサの場合のコードサイズに比べると大きい。

【０１２１】次に、命令長を４０ビット固定としたモデルとの比較を行う。図１２は、図８に示された処理を、命令長が４０ビット固定であり並列実行の境界の情報を命令内に持つ方式のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図である。

【０１２２】このプログラムは、７個の命令で構成されており、命令供給単位としては３個のパケット８０～８２から構成されている。各命令の処理内容は、実行コードの各フィールドに置かれたニーモニックで表現されている。３２ビットの定数のレジスタへの転送についても、１命令で指定することが可能である。

【０１２３】図１２を見ると分かるように、このモデルでは３２ビットの定数のレジスタへの転送を１命令で指定することができるため、実行単位の数は３つであり、実行サイクル数は本発明に係るプロセッサの場合と同じである。

【０１２４】命令数は本発明に係るプロセッサの場合と同じだが、本発明に係るプロセッサの場合は長いビット数を必要としない命令については２１ビット命令で指定できるのに対し、このモデルではすべての命令を４０ビット命令で指定する必要があるため、コードサイズは４０ビット命令が７個で２８０ビットとなっており、本発明に係るプロセッサの場合に比べて大きくなっている。

【０１２５】以上、本発明に係るプロセッサについて、実施形態に基づいて説明したが、本発明はこれらの実施形態に限られないことは勿論である。即ち、

（１）上記実施の形態では、静的なスケジューリングを前提としていたが、本発明はこれに限定されるものではない。つまり、スーパースカラ方式のように動的なスケジューリングを行うプロセッサにも適用することができ

る。この場合は、命令フォーマット内の並列実行境界情報を無くし、解読部の中に動的に並列実行可能か否かを検出する並列実行可否検出装置を持たせ、本実施形態において命令発行制御部にて並列実行境界情報を参照して行っていた制御を、並列実行可否検出装置の出力を参照して行えばよい。このような構成にしても、可変長命令方式においてハードウェアを簡単化できるという本発明の有意性は保たれる。

（２）上記実施の形態では、３個の命令を同時実行するように構成していたが、本発明はこの同時実行命令数に限定されるものではない。例えば、２個の命令を同時発行する構成にしてもよい。この場合は、解読部と命令レジスタ周辺の構成を図１６のブロック図に示すように変更し、実行部の演算器の構成を適宜変更すればよい。

（３）上記実施の形態では、図１の命令フォーマットからわかるように、ユニット１個または２個にて１個の命令を構成していたが、本発明はこのユニット数に限定されるものではない。つまり、３個以上のユニットを連結して１個の命令を構成するような命令フォーマットを定義してもよい。例えば、１～４個の単位命令にて命令を構成する場合には、命令内のフォーマット情報を２ビットにすればよい。

（４）上記実施の形態では、図１の命令フォーマットからわかるように、ユニット１個または２個にて１個の命令を構成していたが、必ずしもユニット単体で構成される命令が存在する必要はない。例えば、１命令が２個または３個のユニットから構成されるとしてもよい。この場合には、命令レジスタと命令デコーダおよび定数オペランドを結ぶ配線を変更すればよい。

（５）上記実施の形態では、図１の命令フォーマットからわかるように、命令内に並列実行の境界であるか否かの情報を持たせていたが、この情報は必ずしも必要ではない。つまり、命令内にはフォーマットに関する情報のみをもち、並列実行可能な命令が存在しない場合には“ｎｏｐ”命令を配置するという方法をとってもよい。この場合においても、各命令の指定に必要な長さの命令フォーマットにて命令を指定することができるという本発明の有意性が保たれる。

（６）上記実施の形態では、図１の命令フォーマットからわかるように、４２ビット命令を構成する２つのユニットのうち２番目のユニットには定数オペランドの一部のみを配置するようになっていたが、このユニットにオペコードを配置しても構わない。そのためには、図５において直接定数オペランドの一部として出力していたユニットを命令デコーダへ入力するように変更し、命令デコーダの入力ビット幅を増加させればよい。

（７）上記実施の形態では、命令バッファの構成として図４に示すものとしたが、本発明はこの構成およびバッファのサイズに限定されるものではない。例えば、一本の単純なキュー構造の命令バッファを用いてもよい。

【０１２６】

【発明の効果】以上の説明から明らかなように、本発明のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあることを特徴とする。

【０１２７】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができ、コードサイズを削減することができる。

【０１２８】また本発明のプロセッサは、一語が複数でかつ所定数の語素からなる複合命令を実行するＶＬＩＷ方式のプロセッサであって、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあり、前記複合命令中の前記単位命令と該単位命令に後続する単位命令とが並列に実行できるか否かを示す並列実行情報が、前記複合命令内に明示的に含まれることを特徴とする。

【０１２９】これによって、各命令の指定に必要なビット数に応じた命令長の命令フォーマットにて命令を指定することができると共に、一般のＶＬＩＷなどで並列実行可能な命令が存在しない場合に挿入される無動作命令の挿入が不要となり、コードサイズを削減することができる。

【０１３０】また本発明のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読出し手段により読出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれ前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記命令単位を組み立てて解読する解読手段とを備え、前記単位命令は、第１の数の前記語素から構成される単位命令と、前記第１の数より大きい第２の数の前記語素から構成される単位命令とがあり、前記第２の数の前記語素から構成される単位命令の内にあって、前記第１の数の前記語素から構成される単位命令に相当する部分からはみ出す部分の語素には、命令中の数値に関する記述のみが配置されることを特徴とする。

【０１３１】これによって、数値に関する記述に関しては命令デコーダに入力せず、直接定数オペランドとして出力することができ、命令デコーダを簡単化することができる。これにより、ハードウェアを単純化することができる。

【０１３２】また本発明のプロセッサは、一語が複数の語素からなる複合命令を実行し、実行の単位となる単位

命令が一または複数の前記語素から構成され、かつ前記単位命令を構成する前記語素の数が少なくとも二通りあるプロセッサであって、記憶装置から前記複合命令を読み出す複合命令読み出し手段と、前記複合命令読出し手段により読出された前記複合命令中の前記語素から、いずれかの前記語素内に含まれ前記単位命令を構成する前記語素の数に関する語数情報に基づいて、前記命令単位を組み立てて解読する解読手段とを備え、前記解読手段は、前記語素の一定数ごとに区切られた前記複合命令をそれぞれ解読する複数の単位解読手段と、前記語数情報に基づいて前記単位解読手段のいくつかの解読を無効化する解読制御手段とから構成されることを特徴とする。

【０１５５】これによって、命令の供給部から命令の発行部への単位命令の転送の際のセレクタによる遅延を解消し、またハードウェアを単純化することができる。よって、命令長を可変とすることによりハードウェアが複雑になり、遅延が増大して性能が低下する、という問題点を克服している。さらに、この命令発行制御方法では、命令長を増して同時発行可能な命令数が増えても、単位命令からレジスタへの転送の遅延が増加しないというメリットがある。

【０１５６】さらに、ＧＭＩＣＲＯ／４００で採用されている構成に対してデコーダの数を減らすことができる。また、解読ステージの前半にオペランドの読み出しを開始することができるため、解読ステージが完了する時点でオペランドの読み出しを完了させておくことができ、実行効率を高めることができる。

【図面の簡単な説明】

【図１】本発明の実施形態に係るプロセッサが実行する命令の構成を示す図

【図２】同プロセッサにおける命令の供給と発行の概念を示す図

【図３】同プロセッサのハードウェア構成を示すブロック図

【図４】同プロセッサの命令バッファ２２の詳細な構成を示すブロック図

【図５】同プロセッサの命令レジスタ２３周辺の構成を示すブロック図

【図６】同プロセッサの命令発行制御部３１とその周辺の回路構成を示す図

【図７】同プロセッサが同時発行可能な命令群の命令長の組み合わせを示す図

【図８】３２ビットの定数を扱う処理の一例を示すフローチャート

【図９】図８に示された処理を図３のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図１０】図８に示された処理を命令長が３２ビット固定のＶＬＩＷ方式の従来のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図１１】図８に示された処理を、命令長３２ビット固定で命令内に並列実行境界の情報を持たせる方式の従来のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図１２】図８に示された処理を、命令長４０ビット固定で命令内に並列実行境界の情報を持たせる方式の従来のプロセッサに行わせるプログラムの実行コードの例と実行イメージを示す図

【図１３】従来のプロセッサにおける命令レジスタ周辺の構成を示すブロック図

【図１４】従来のプロセッサにおける命令レジスタ周辺の構成を示すブロック図

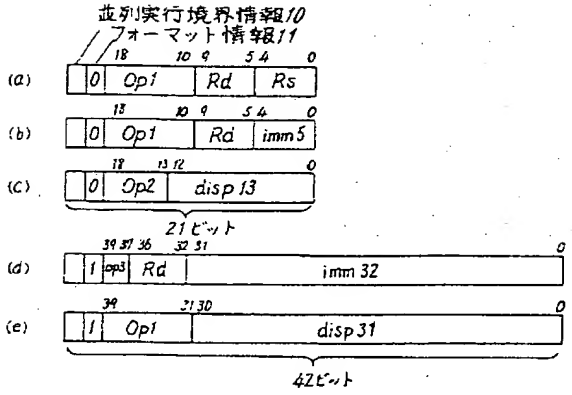
【図１５】従来のプロセッサの一例であるＧＭＩＣＲＯ／４００における命令レジスタ周辺の構成を示すブロック図

【図１６】本発明の別の実施形態のプロセッサにおける命令レジスタ２３周辺の構成を示すブロック図

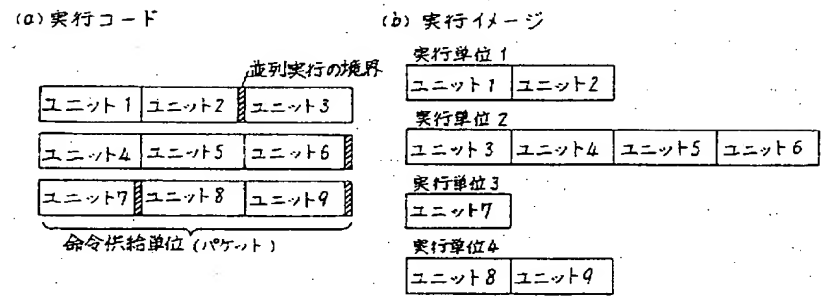
【符号の説明】

１０　並列実行境界情報
１１　フォーマット情報
２０　命令供給発行部
２１　命令フェッチ部
２２　命令バッファ
２３　命令レジスタ
３０　解読部
３１　命令発行制御部
３２　命令デコーダ
３３　第１命令デコーダ
３４　第２命令デコーダ
３５　第３命令デコーダ
４０　実行部
４１　実行制御部
４２　ＰＣ部
４３　レジスタファイル
４４　第１演算部
４５　第２演算部
４６　第３演算部
４７　オペランドアクセス部
４８、４９　データバス
５０　ユニットキュー
２２１　命令バッファＡ
２２２　命令バッファＢ
２２３　命令バッファ制御部
２２４ａ～２２４ｄ　セレクタ
２３１　命令レジスタＡ
２３２　命令レジスタＢ
２３３　命令レジスタＣ
２３４　命令レジスタＤ

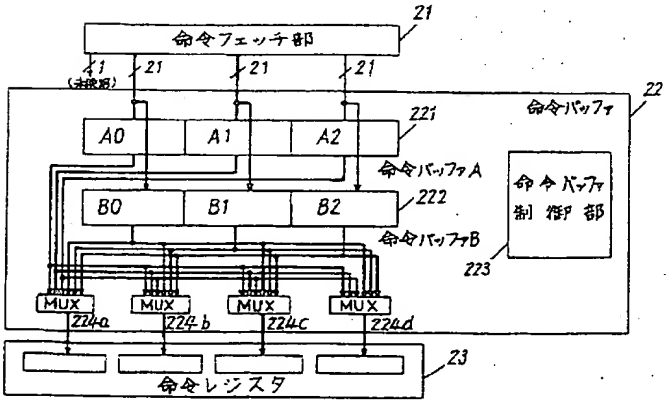
【図1】


並列実行境界情報10
フォーマット情報11
(a)
(b)
(c)
(d)
(e)
Op1
Rd
Rs
imm5
Op2
disp13
21ビット
imm32
disp31
42ビット

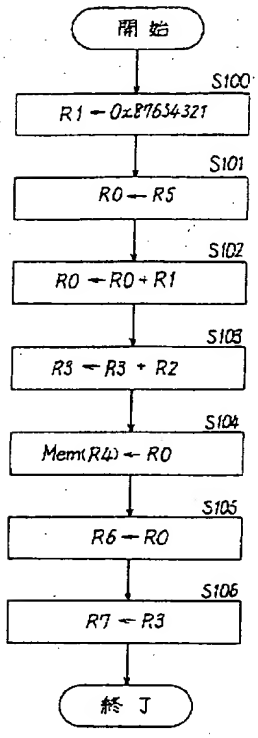
【図2】


(a) 実行コード
(b) 実行イメージ
並列実行の境界
ユニット1
ユニット2
ユニット3
ユニット4
ユニット5
ユニット6
ユニット7
ユニット8
ユニット9
命令供給単位（パケット）
実行単位1
実行単位2
実行単位3
実行単位4

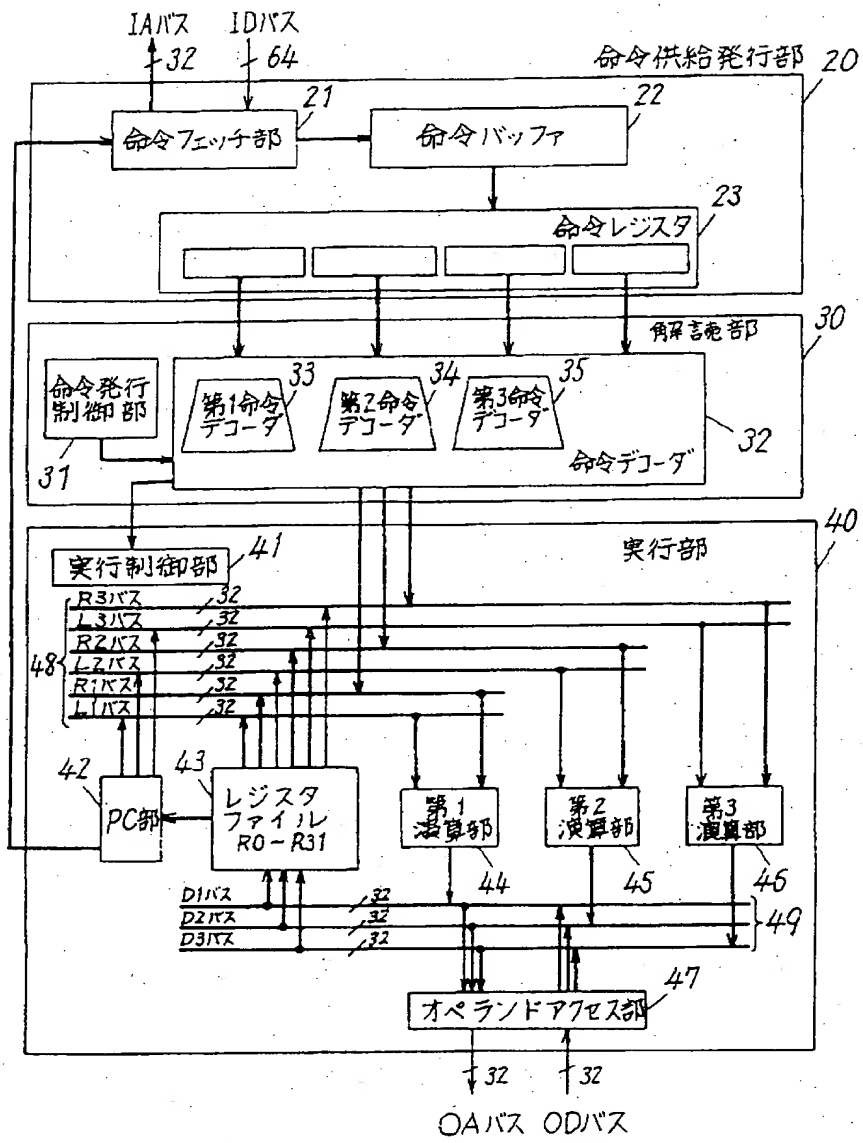
【図4】


命令フェッチ部
21
命令バッファ
22
A0
A1
A2
221
命令バッファA
B0
B1
B2
222
命令バッファB
命令バッファ制御部
223
MUX
224a
224b
224c
224d
命令レジスタ
23

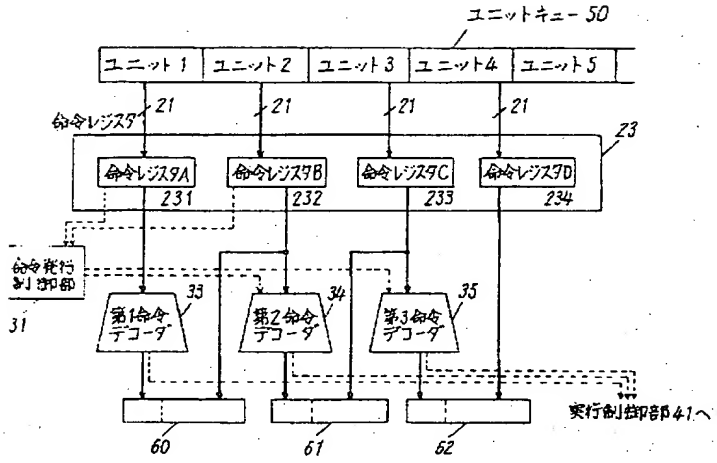
【図8】


開始
S100
R1←0x87654321
S101
R0←R5
S102
R0←R0+R1
S103
R3←R3+R2
S104
Mem(R4)←R0
S105
R6←R0
S106
R7←R3
終了

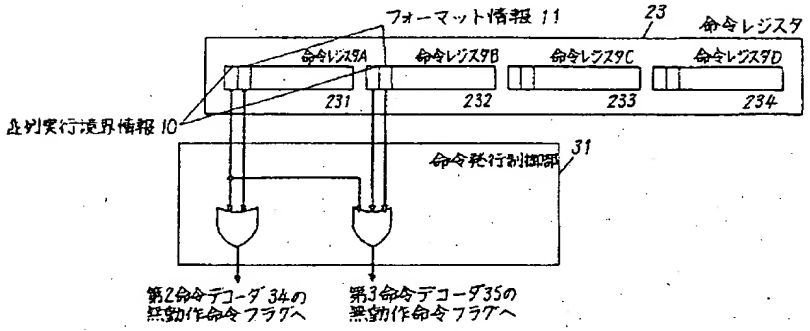
【図3】


IAバス
32
IDバス
64
命令供給発行部
20
21
命令フェッチ部
22
命令バッファ
23
命令レジスタ
解読部
30
命令発行制御部
31
第1命令デコーダ
33
第2命令デコーダ
34
第3命令デコーダ
35
命令デコーダ
32
実行部
40
実行制御部
41
R3バス
L3バス
R2バス
L2バス
R1バス
L1バス
48
42
PC部
43
レジスタファイル
R0－R31
第1演算部
第2演算部
第3演算部
44
45
46
D1バス
D2バス
D3バス
49
47
オペランドアクセス部
OAバス
ODバス

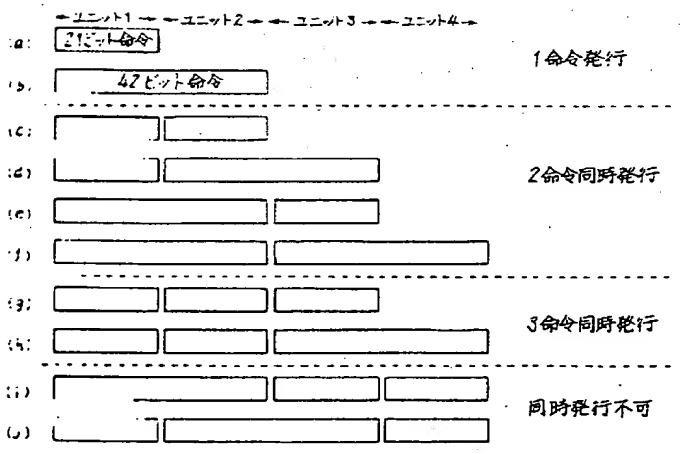
【図5】


ユニットキュー 50
ユニット1
ユニット2
ユニット3
ユニット4
ユニット5
21
命令レジスタ
23
命令レジスタA
231
命令レジスタB
232
命令レジスタC
233
命令レジスタD
234
命令発行制御部
31
第1命令デコーダ
33
第2命令デコーダ
34
第3命令デコーダ
35
実行制御部41へ
60
61
62

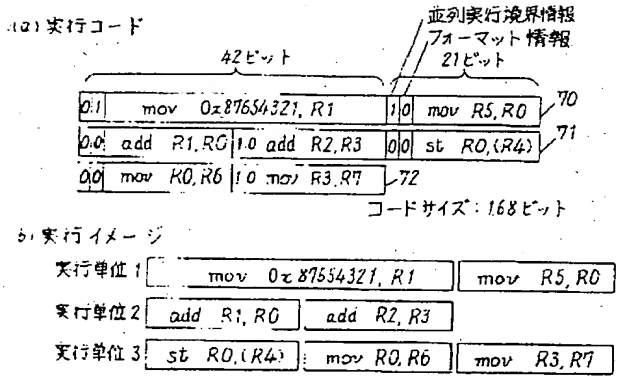
【図6】


フォーマット情報 11
23
命令レジスタ
命令レジスタA
命令レジスタB
命令レジスタC
命令レジスタD
231
232
233
234
並列実行境界情報 10
命令発行制御部
31
第2命令デコーダ 34の無効作命令フラグへ
第3命令デコーダ35の無効作命令フラグへ

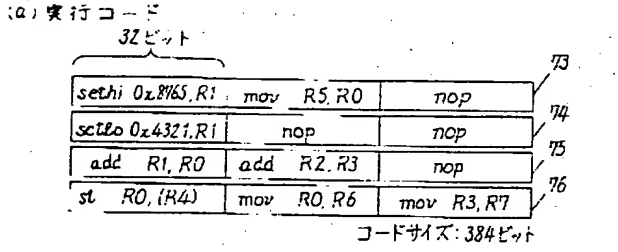
【図7】


ユニット1
ユニット2
ユニット3
ユニット4
(a)
21ビット命令
1命令発行
(b)
42ビット命令
(c)
(d)
2命令同時発行
(e)
(f)
(g)
(h)
3命令同時発行
(i)
(j)
同時発行不可

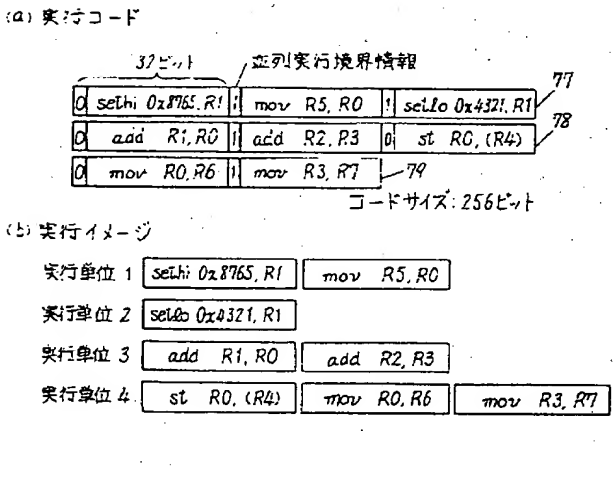
【図9】



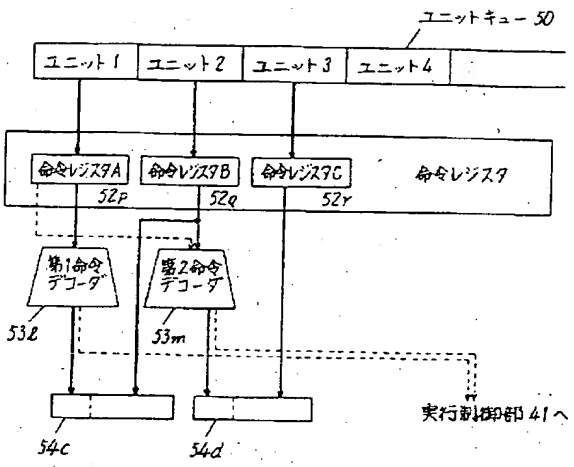
【図10】



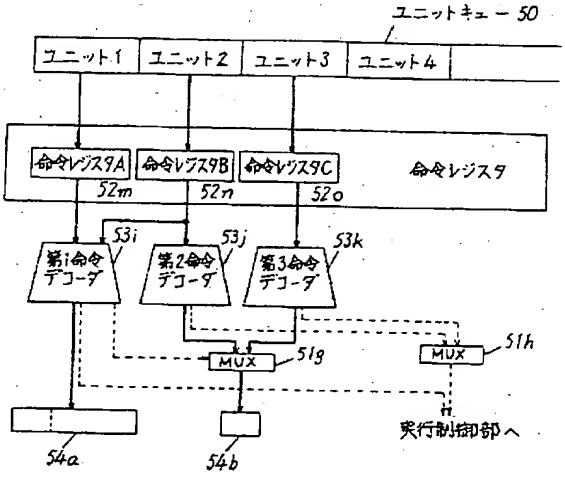
【図11】



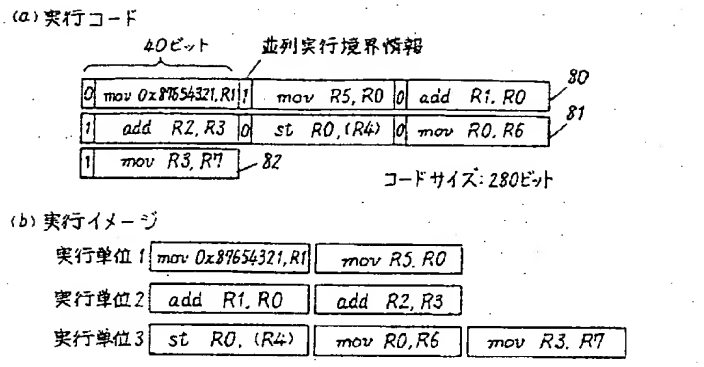
【図16】



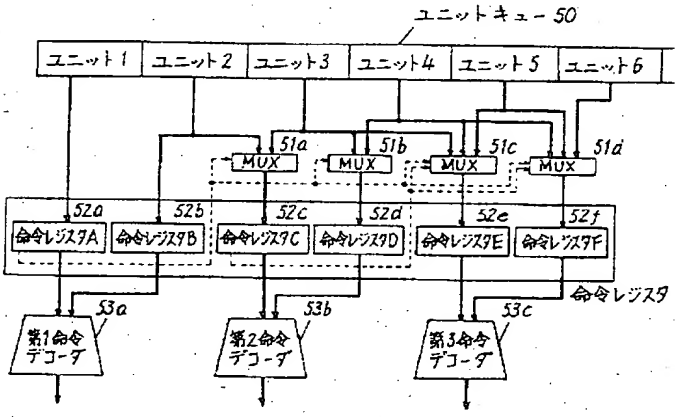
【図15】



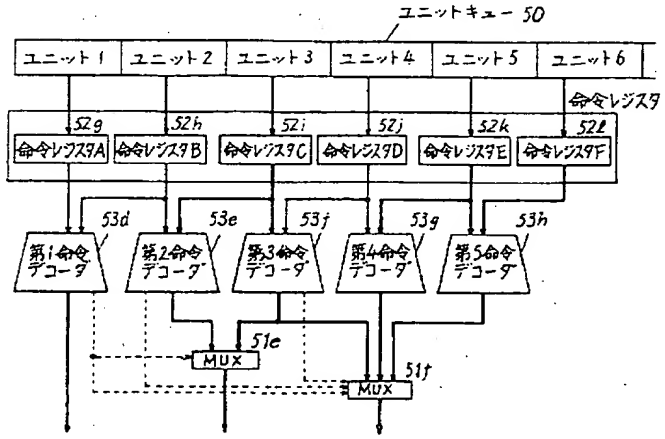
【図１２】


(a) 実行コード
40ビット
並列実行境界情報
0 mov 0x87654321,R1
1 mov R5, R0
0 add R1, R0
80
1 add R2, R3
0 st R0,(R4)
0 mov R0, R6
81
1 mov R3, R7
82
コードサイズ: 280ビット
(b) 実行イメージ
実行単位1
mov 0x87654321,R1
mov R5, R0
実行単位2
add R1, R0
add R2, R3
実行単位3
st R0, (R4)
mov R0, R6
mov R3, R7

【図１３】


ユニットキュー 50
ユニット1
ユニット2
ユニット3
ユニット4
ユニット5
ユニット6
51a
51b
51c
51d
MUX
52a
52b
52c
52d
52e
52f
命令レジスタA
命令レジスタB
命令レジスタC
命令レジスタD
命令レジスタE
命令レジスタF
命令レジスタ
53a
53b
53c
第1命令デコーダ
第2命令デコーダ
第3命令デコーダ

【図１４】


ユニットキュー 50
ユニット1
ユニット2
ユニット3
ユニット4
ユニット5
ユニット6
命令レジスタ
52g
52h
52i
52j
52k
52l
命令レジスタA
命令レジスタB
命令レジスタC
命令レジスタD
命令レジスタE
命令レジスタF
53d
53e
53f
53g
53h
第1命令デコーダ
第2命令デコーダ
第3命令デコーダ
第4命令デコーダ
第5命令デコーダ
51e
51f
MUX

フロントページの続き

(72)発明者　高山　秀一
大阪府門真市大字門真1006番地　松下電器産業株式会社内

(72)発明者　小谷　謙介
大阪府門真市大字門真1006番地　松下電器産業株式会社内